



Cahier des charges

Amélioration d'un programme de cartographie par séquençage: création d'une interface d'entrée/sortie.

Auteurs :

Hermes PARAQUINDES
Juliette GEOFFRAY
Eric CUMUNEL

Encadrants :

Fabrice BESNARD
Laurent GUEGUEN

12 Février 2018

Contents

1	Présentation du projet	2
1.1	Contexte	2
1.2	Objectifs	2
1.3	Description de l'existant	3
1.4	Critères d'acceptabilité du produit	4
2	Expression des besoins	5
2.1	Besoins fonctionnels	5
2.1.1	Fonctionnalité du logiciel	6
2.1.2	Fonctionnalité du code	6
2.2	Besoins non fonctionnels	6
3	Contraintes	6
3.1	Délais	6
3.2	Autres contraintes	7
4	Déroulement du projet	7
4.1	Planification	7
4.2	Plan d'assurance qualité	7
4.3	Responsabilités	7
4.3.1	Maîtrise d'ouvrage	7
4.3.2	Maîtrise d'œuvre	7
5	Bibliographie	8

1 Présentation du projet

1.1 Contexte

La mutagenèse est un processus par lequel l'information génétique d'un organisme, et donc celle de son ADN est modifiée, ce qui entraîne une mutation.

L'apparition des mutants dans une population est un phénomène rare et donc avec une très faible probabilité de se produire. Afin d'augmenter cette probabilité, les organismes peuvent être traités par des agents mutagènes qui vont introduire des mutations (remplacement, modification ou endommagement) sur l'organisme de manière aléatoire. Ces agents mutagènes peuvent être de nature chimique comme EMS (Méthanesulfonate d'éthyle), de nature physique comme la lumière ultraviolette et les radiations ionisantes ou de nature bactérienne pathogénique contenant des plasmides (ex: T DNA) capables de d'intégrer au génome de l'hôte une séquence ADN (capable de s'exprimer ou non).

Provoquer des mutations dans un organisme d'intérêt permet de localiser les gènes d'intérêts, de les cartographier et d'en déduire des informations sur le rôle des gènes. L'identification des mutations responsables du phénotype du mutant constitue le principe de base de la génétique mendélienne. La méthode utilisée pour identifier la mutation recherchée est celle du clonage positionnel.

C'est une méthode utilisée dans les cas où on ne connaît ni la séquence, ni la fonction du gène mais dont la mutation est supposée être à l'origine d'un caractère phénotypique visible. Un croisement avec une lignée génétiquement différente n'ayant pas le phénotype mutant est réalisé. Suite au croisement, il est nécessaire de génotyper un grand nombre d'individu et cela est une tâche fastidieuse prenant beaucoup de temps. La révolution dans les nouvelles technologies de séquençage et d'assemblage du génome a facilité le processus. Aujourd'hui on peut séquencer plusieurs mutants en même temps et analyser les variations génomiques sur tout le génome en une seule fois.

Toutefois, c'est une analyse bio-informatique qui requiert de nombreuses étapes ainsi que l'utilisation de plusieurs programmes et logiciels distincts. La plupart des programmes utilisés sont adaptés à un organisme modèle ou à un « design » génétique, et dépendent de serveurs distants. Afin de faciliter l'étape de cartographie et l'identification des mutations, un pipeline appelé *Andalusian_Mapping* a été développé. Ce pipeline permet de travailler avec différentes espèces et souches de cartographie.

1.2 Objectifs

L'objectif général de ce projet est de rendre *Andalusian_Mapping* plus accessible à la communauté scientifique, notamment pour les biologistes non-informaticiens, à la fois au niveau de la mise en place des outils du pipeline que de l'affichage des résultats.

Pour ce faire, plusieurs pistes peuvent se révéler intéressantes :

- Au niveau de la mise en place : incorporer le programme dans un docker contenant les dépendances logicielles nécessaires
- Au niveau de l'importation des données de départ : intégrer une interface graphique permettant d'ajouter les fichiers plus aisément
- Au niveau de l'exploitation des résultats finaux : créer une interface graphique et/ou une page HTML pour visualiser les différents résultats (sous la forme de graphes et de tableau)
- Améliorer l'organisation du pipeline (optionnel)

1.3 Description de l'existant

Récemment, Cloudmap, un pipeline automatique de mapping-by-sequencing et d'identification de mutants a été développé et intégré à Galaxy, qui est une interface utilisateur simple et intuitive. Cependant, ce pipeline ne permet pas à ce jour de travailler avec des génomes de référence *exotiques*, étant donné que seuls les génomes modèles comme *Caenorhabditis elegans* ou *Arabidopsis thaliana* sont disponibles sur Galaxy. *Andalusian Mapping* a donc été développé dans le but de fournir un outil similaire afin de travailler avec une gamme plus vaste d'organisme.

Andalusian Mapping est un pipeline permettant d'effectuer une cartographie par séquençage afin d'identifier les régions les plus susceptibles d'être responsables d'une mutation. Ce pipeline est un script bash permettant l'utilisation en une seule étape de plusieurs outils bioinformatiques.

Il tourne sous environnement Linux et Mac. La mise en place de différents logiciels est cependant fastidieuse et nécessaire au fonctionnement de ce pipeline.

Logiciels requis :

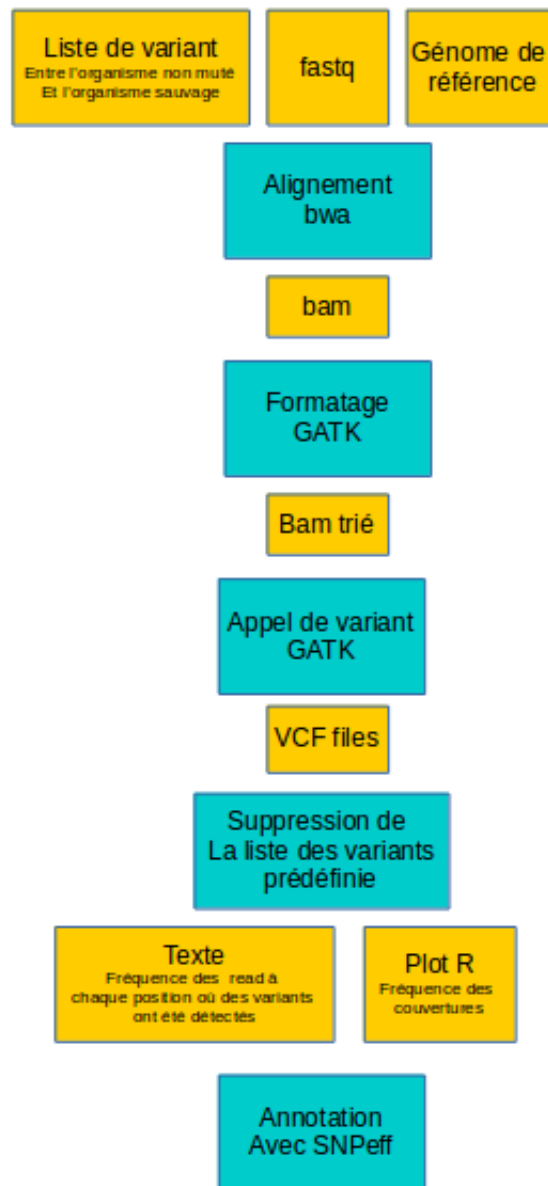
- bwa version 0.7.5a-r405 ou supérieur
- samtools version 0.1.18 ou supérieur
- Picard Version 1.110 ou supérieur
- GATK Version 3.7 ou supérieur
- R Version 3.3 ou supérieur, avec le package ggplot2.
- snpEff Version 4.1g ou supérieur

L'utilisation de ce pipeline se base sur un fichier de configuration à remplir préalablement contenant les chemins vers les exécutables des logiciels utilisés, les fichiers *fastq*, le génome de référence, une liste de variants prédéfinie ainsi qu'un répertoire dans lequel seront envoyés les résultats intermédiaires et finaux.

Description du pipeline dans son ensemble :

1. La première étape consiste à aligner les reads contre un génome de référence et à trier le fichier de sortie (.bam) pour les besoins d'un logiciel intervenant plus tard (PICARD).
2. Ensuite on ajoute des informations de groupes aux reads.
3. L'étape suivante consiste au merge des séquences et la suppression des reads dupliqués.
4. On indexe le fichier bam pour les besoins d'un logiciel intervenant plus tard (picard).
5. Génération de statistiques.
6. On réaligne les reads en prenant en compte les informations récoltées jusqu'ici dans le pipeline.
7. On réalise une étape de Base Quality Score Recalibration avec génération de graphique et de statistique de qualité.
8. On réalise l'appel de variant.

9. On va réaliser la cartographie de la mutation et on calcul la fréquence de l'allèle JU170 pour chaque scaffold.
10. On va trouver les meilleurs locus candidats grâce au annotation.
11. On va générer des graphiques a l'aide de R et un fichier texte comprenant les locus responsable de la mutation les plus probables.



1.4 Critères d'acceptabilité du produit

Afin de répondre à la problématique, une interface graphique sera développée. Elle devra permettre la rédaction du formulaire en entrée de pipeline, son exécution et enfin, la visualisation des résultats de sorties.

Le logiciel devra être simple d'utilisation et les résultats facilement interprétables.

2 Expression des besoins

2.1 Besoins fonctionnels

2.1.1 Fonctionnalité du logiciel

Le logiciel devra remplir la même fonction que précédemment, tout en présentant une mise en place simplifiée et une visualisation des résultats améliorée, afin de permettre à des utilisateurs possédant peu de connaissances en informatique de l'utiliser.

2.1.2 Fonctionnalité du code

Les fonctionnalités que doit remplir le code permettant la rédaction du fichier d'entrée du pipeline:

- remplir les champs du fichier source (configuration du pipeline) :
 - Génome de référence
 - Fastq en pairend
 - Liste d'allèles connus entre l'individu sauvage et l'individu pas encore muté. Pour pouvoir les supprimer lors de l'appel de variant entre l'organisme sauvage et l'organisme muté.
 - Données d'information sur les reads (comme la technique de séquençage utilisée ex: Illumina, le SNP utilisé pour repérer la région mutée...)
 - Le nom de sortie du dossier et de l'extension ajoutée au fichier.
- Dans un second temps, on pourra prévoir un remplissage de certains champs automatiquement via des pipelines développés par le maître d'ouvrage. Comme la génération de la liste des variant entre l'organisme sauvage et l'organisme pas encore muté.

Les fonctionnalités que doit remplir le code permettant la sortie des résultats:

- Visualisation claire des données et des graphiques d'intérêt.
- Un fichier de sortie exportable comme une page html, pour faciliter le partage de ces résultats.

2.2 Besoins non fonctionnels

L'interface développée est réalisée sous le système d'exploitation Linux, il comprendra des parties en Shell, en Python et on utilisera également *pyQt*, une librairie graphique python.

3 Contraintes

3.1 Délais

Le projet s'organise sur une période de 4 semaines. Il débutera le lundi 5 février et prendra fin le vendredi 2 mars. Le projet pourra être continué durant une période de cours allant du 5 au 29 mars.

Un cahier des charges définitif devra être présenté le lundi 12 février.

Le livrable final devra être rendu le 29 mars 2018.

3.2 Autres contraintes

4 Déroulement du projet

4.1 Planification

Listes des tâches :	Distribution des rôles			5 au 9	12 au 16	19 au 23	26 au 2	5 au 29
	Juliette	Hermes	Eric	Février – Mars				
Rendez-vous avec le référent								
Rédaction du cahier des charges								
- Introduction								
- Existant								
- Autre partie								
Installation et vérification de la bonne marche du pipeline								
Réécriture du Pipeline								
- Ajout de condition a chaque création de fichier								
- Vérification des fichiers créer								
Mise en place d'un docker								
- Lecture de la documentation								
- Mise en place du docker								
Interface graphique								
Développement de la partie formulaire								
- Formulaire simple								
- Formulaire interactif								
Développement de la partie recharge d'un formulaire								
Développement de la partie exécution du pipeline								
Développement de la partie visualisation des résultats								
Développement de la partie recharge d'ancien résultats								
Beta testing								
Documentation								

4.2 Plan d'assurance qualité

Le contrôle qualité de l'interface développée consistera en une bonne exécution, simple et sans erreurs. Le logiciel sera testé par des membres de différentes équipes :

- Validation par Dr Besnard
- Utilisation du logiciel par des utilisateurs non initiés en informatique

4.3 Responsabilités

4.3.1 Maîtrise d'ouvrage

Ce projet nous a été proposé par Fabrice Besnard, biologiste rattaché à l'École Normale Supérieure de Lyon (ENS), travaillant au Laboratoire de Reproduction et Développement des Plantes (RDP).

4.3.2 Maîtrise d'œuvre

Pour réaliser ce projet, nous serons trois étudiants en Master 1 de Bio-informatique à Lyon 1 : Juliette Geoffray, Hermes Paraquindes et Eric Cumunel.

5 Bibliographie

1. Schneeberger, K. Using next-generation sequencing to isolate mutant genes from forward genetic screens. *Nat. Rev. Genet.* 15, 662–676 (2014).
2. Galaxy | Published Page | CloudMap. Available at: <https://usegalaxy.org/u/gm2123/p/cloudmap>. (Accessed: 5th September 2017)
3. Home of SHOREmap. Available at: <http://bioinfo.mpipz.mpg.de/shoremap/index.html>. (Accessed: 5th September 2017)
4. MutMap - Genomics & Breeding. Available at: <http://genome-e.ibrc.or.jp/home/bioinformatics-team/mutmap>. (Accessed: 5th September 2017)
5. Besnard, F., Koutsovoulos, G., Dieudonné, S., Blaxter, M. & Félix, M.-A. Toward Universal Forward Genetics: Using a Draft Genome Sequence of the Nematode *Oscheius tipulae* To Identify Mutations Affecting Vulva Development. *Genetics* 206, 1747–1761 (2017).
6. Besnard, F. Andalusian_Mapping: Scripts to perform mapping-by-sequencing. (2017).